

On Finding Ordinary or Monochromatic Intersection Points*

George B. Purdy[†]

Justin W. Smith[‡]

October 1, 2009

Abstract

An algorithm is demonstrated that finds an ordinary intersection in an arrangement of n lines in \mathbb{R}^2 , not all parallel and not all passing through a common point, in time $O(n \log n)$. The algorithm is then extended to find an ordinary intersection among an arrangement of hyperplanes in \mathbb{R}^d , no d passing through a line and not all passing through the same point, again, in time $O(n \log n)$.

Two additional algorithms are provided that find an ordinary or monochromatic intersection, respectively, in an arrangement of pseudolines in time $O(n^2)$.

1 Introduction

Over a century ago Sylvester posed the question of whether a set of n non-collinear points necessarily determine an ordinary line [21]. (An *ordinary* line

*AMS MSC-2000: 51A99, 68U05. ACM CCS-1998: I.3.5.

[†]george.purdy@uc.edu. Department Of Computer Science, 814 Rhodes Hall, Cincinnati, OH 45221

[‡]smith5jw@email.uc.edu. Department Of Computer Science, 814 Rhodes Hall, Cincinnati, OH 45221

is one incident to exactly two points.) Although it was thought to be true, no proof was found until the problem was raised again by Erdős in the 1930's. Soon after, it was proven by Gallai and his proof was published in [20]. Hence, it is now called the Sylvester-Gallai Theorem. (See also [3] for an elegant proof by L. M. Kelly.)

Since Sylvester originally posed his question in 1893, a variety of related questions have been asked. One well known variation relates to a two-colored, or bichromatic, set of points. Ron Graham first asked (around 1965, see [8]) whether a bichromatic set of non-collinear points necessarily determines a monochromatic line, i.e., a line determined by two or more points all of which are the same color. The first published proof was a few years later by Chakerian [2]. Earlier than Chakerian (and referenced in his paper), Motzkin and Rabin had proofs of this result in its dual form. (Motzkin's proof was published in [6]¹.) This theorem is now commonly called the Motzkin-Rabin Theorem.

The algorithms in this article deal with arrangements of hyperplanes (i.e., $(d - 1)$ -flats in \mathbb{R}^d) or pseudolines in the euclidean plane. By duality, the algorithms on hyperplanes can be used, as well, on a point configuration to solve the dual problem. However, hyperplane arrangements are more general than a dual of points, e.g., any two points determine a line, but two parallel lines do not determine an intersection point. Thus, some problem instances (i.e., those involving parallel hyperplanes) can only be solved by algorithms that work in the domain of hyperplane arrangements.

The first algorithm presented finds an ordinary intersection point in an arrangement of lines (some possibly parallel) in \mathbb{R}^2 in time $O(n \log n)$. This algorithm will subsequently be used to solve the same problem for hyperplanes in \mathbb{R}^d .

¹Grünbaum states in [8] that the proof published in [6] is actually due to Motzkin, although the text attributes it to S. K. Stein.

2 Ordinary Points in an Arrangement of Lines in \mathbb{R}^2

2.1 Existence of Ordinary Intersection Points

Dirac conjectured in 1951 that in any set of $2n$ points, there exist n ordinary lines [5]. The best known lower bound is $\frac{6n}{13}$ ordinary lines in a set of n points, found by Csimá and Sawyer in [4]. This improved upon the Kelly and Moser result of $\frac{3n}{7}$ [10].

Since an ordinary line always exists among a set of non-collinear points, an obvious question within computational geometry is how to find one. A naive method would potentially take time $O(n^3)$ by considering for each point pair whether a third point is collinear. Mukhopadhyay et al. improved this by finding an algorithm that finds an ordinary line among a set of points in time $O(n \log n)$ [15]. A similar, but simplified, algorithm was demonstrated several years later by Mukhopadhyay and Green [16].

In [11], Lenchner considers the “sharp dual” of this problem, i.e., ordinary intersections determined by an arrangement of lines in \mathbb{R}^2 , not all parallel and not all passing through a common point. (Since the “sharp dual” is more general than the “dual”, the Csimá and Sawyer result does not apply.) Lenchner first proved that ordinary intersections occur in such an arrangement, and in fact, that there must exist at least $\frac{5n}{39}$ such points among n lines. He later improved this original result to $\frac{2n-3}{7}$ among $n \geq 7$ lines [12].

In the conclusion of [11], Lenchner asks whether an algorithm exists that can find an ordinary intersection in such an arrangement in time $o(n^2)$. The following algorithm performs in time $O(n \log n)$.

2.2 Locating an Ordinary Intersection

Definition 2.1. Let L_0 , L_1 , and L_2 be any three lines of \mathcal{A} that intersect at three distinct points. Label the lines such that L_0 and L_1 intersect at point P , which is to the left of the intersection Q of lines L_0 and L_1 . Points P and Q are *consecutive points* (on L_0) if no line intersects L_0 on the open interval (P, Q) . Furthermore, if P and Q are consecutive points, L_1 and L_2 are *consecutive lines* (with respect to L_0) if L_1 is the “rightmost” line through P and L_2 is the “leftmost” line through Q . In other words, there is no line through P intersecting L_2 at a point closer to Q than $L_1 \cap L_2$, and there is no line through Q closer to P than $L_1 \cap L_2$.

Lemma 2.2. *Suppose line L_0 contains no ordinary points. Let X be the closest intersection point above line L_0 incident to at least two lines not parallel to L_0 . Then, X must be the intersection of two consecutive lines through two consecutive points P_i and P_{i+1} .*

Proof. Suppose X is the intersection of lines, L_1 and L_2 , through consecutive points, P_i and P_{i+1} , but the lines are not consecutive. Thus, there exists a line through either P or Q that intersects either L_1 and L_2 at a point closer than X , i.e., a contradiction

So, suppose there are three intersection points, P , Q , and R on L_0 in that order from left to right, and X is the intersection of a line through P and a line through R . Then there is another line through Q that intersects one side of the triangle $\triangle XPR$, interior to the side PX or RX . Either way, there is an intersection point S that is lower than X , i.e., a contradiction. \square

Lemma 2.3. *Suppose line L_0 contains no ordinary points, and X is the closest intersection point above L_0 . Then X is either an ordinary point, or it has a line M through it parallel to L_0 .*

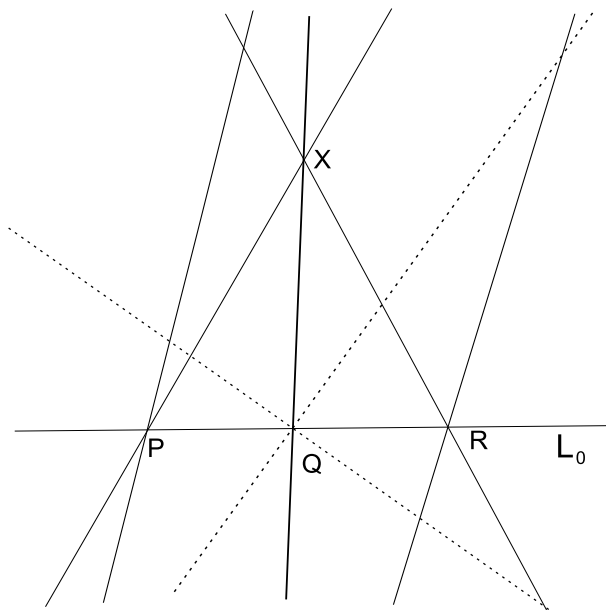


Figure 1: A line passing through (non-ordinary) point Q will determine an intersection closer to L_0 than X .

Proof. Let P and Q be points in order from left to right on L_0 that contain the lines forming X . (By Lemma 2.2, P and Q are consecutive points and X is formed by consecutive lines through these.) Suppose that there is a third line \overline{XR} through X , where R is another point on L_0 . Without loss of generality let's suppose that R is to the right of Q . There is another line through Q that intersects either segment PX or segment RX , and either way X is not the lowest point, i.e., a contradiction. See Figure 1. \square

Together, Lemmas 2.2 and 2.3 can be used to prove the dual form of the Sylvester-Gallai Theorem. (A configuration of points can always be dualized such that no two lines are parallel.) From the algorithm below, one can also see proof of its “sharp dual” form.

2.3 Algorithm to Find an Ordinary Point in Time $O(n \log n)$

Theorem 2.4. *Given an arrangement of n lines in \mathbb{R}^2 , not all parallel and not all passing through the same point, there exists an algorithm to find a ordinary intersection point in time $O(n \log n)$.*

Proof. Let n be the number of lines in arrangement \mathcal{A} . Let L_0 , L_1 , and L_2 be any three lines of \mathcal{A} that intersect at three distinct points. If no such lines exist then the arrangement consists of only two families of parallel lines and every intersection is ordinary, so suppose this not to be the case. *Time to find L_0 , L_1 , and L_2 : $O(n)$.*

Consider L_0 to be horizontal, and L_1 and L_2 to be intersecting “above” L_0 . Find all of the intersection points on L_0 . Label them from left to right P_1, P_2, \dots, P_m . *Time to sort them, collecting potentially multiple lines into each P_k : $O(n \log n)$.*

If any P_k is ordinary the algorithm is done, so suppose that none are.

Find the “leftmost” and “rightmost” line through each P_k , i.e., find the pairs of consecutive lines. *Time: $O(\deg(P_1) + \deg(P_2) + \dots + \deg(P_m)) = O(n)$.*

Let X be the lowest intersection point above L_0 , which by Lemma 2.2 must be the intersection of consecutive lines through consecutive bundles P_k and P_{k+1} . *Time: $O(n)$.*

Determine whether there is a line M parallel to L_0 that passes through X , and if so, then find M . *Time: $O(n)$.* If there is no such line M then, by Lemma 2.3, X is an ordinary point.

Otherwise, suppose that M exists. Let Y be the intersection of M with the leftmost line from the leftmost bundle P_1 .

Assume Y is not an ordinary point. Then, there exists a point P_k , $k > 1$, such that YP_k is a line of the arrangement, and there is another line through P_1 that intersects YP_k in its interior at a point lower than Y and therefore lower

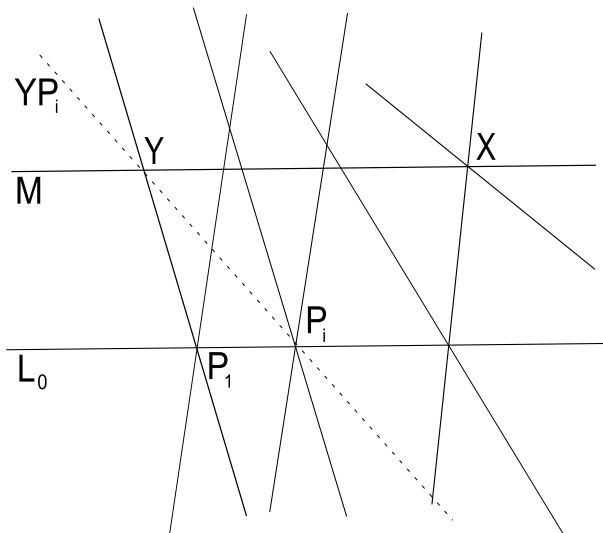


Figure 2: If X is the lowest intersection determined by lines not parallel to L_0 , then Y must be ordinary.

than X , i.e., a contradiction. Hence Y is an ordinary point. (See Figure 2.)

Time to find Y : $O(1)$.

□

3 Ordinary Points in an Arrangement of Hyperplanes in \mathbb{R}^d

3.1 Duality

Given n points in \mathbb{R}^3 , no three on a line and not all on a plane, does there necessarily exist a three-point plane? Recently, the present authors proved that, under the same hypothesis, there must exist at least $\frac{4}{13} \binom{n}{2}$ such planes [19]. Without the assumption that no three points are collinear, Bonnice and Kelly showed that there must exist at least $\frac{3n}{11}$ ordinary planes in 3-space [1]. The existence of such a plane also follows from Hansen's result on the existence of *ordinary* hyperplanes in d -dimensional projective space [9]. (In 3-space their

existence was proved earlier by Motzkin, but for 4 or higher it's due to Hansen.) An ordinary hyperplane in d -space is one in which all but one of the points lie on a $(d - 2)$ -flat.

The algorithm presented in Section 3.2 solves the following problem. Find the intersection point of exactly d hyperplanes in \mathbb{R}^d (i.e., an ordinary intersection) in an arrangement of hyperplanes, not all parallel, not all passing through the same point, and no d passing through a line. This problem on hyperplanes is the “sharp dual” of a problem on points. That is, given a set of n points in \mathbb{R}^d , no d on a $(d - 2)$ -flat and not all on a hyperplane, find a hyperplane determined by exactly d points. (It follows from this hypothesis that no k points lie on a $(k - 2)$ -flat for $3 \leq k \leq d$.)

For the convenience of the reader, we provide the following correspondences between flats and their duals:

- Hyperplanes \longleftrightarrow Points
- $(d - 2)$ -flats \longleftrightarrow Lines
- k -flats $\longleftrightarrow (d - k - 1)$ -flats

3.2 Algorithm to Find an Ordinary Point in Time $O(n \log n)$

Before we claim to have an algorithm to find an ordinary intersection, we must first be sure that a given set of hyperplanes determines an intersection point. Let h_i^\perp be a normal vector to the hyperplane h_i . The following lemma shows the necessary and sufficient conditions for a general intersection point to exist.

Lemma 3.1. *Let $H = \{h_0, h_1, \dots, h_{d-1}\}$ be a set of d hyperplanes in \mathbb{R}^d . The hyperplanes of H determine an intersection point if and only if their normals form a basis for \mathbb{R}^d , i.e., $\text{span}(h_0^\perp, h_1^\perp, \dots, h_{d-1}^\perp) = \mathbb{R}^d$.*

Proof. This follows from the observation that the space orthogonal to the intersection of hyperplanes is the span of the hyperplanes' normals. That is, given a set of k hyperplanes $H' = \{h'_0, h'_1, \dots, h'_{k-1}\}$, then $(h'_0 \cap h'_1 \cap \dots \cap h'_{k-1})^\perp = \text{span}(h'^\perp_0, h'^\perp_1, \dots, h'^\perp_{k-1})$. (Note that the sum of the dimension of a space and the dimension of its orthogonal space in \mathbb{R}^d is always d). \square

Given a set of k vectors one can find a maximal linearly independent subset in time $O(k^2)$ using a method such as row reduction on a matrix. By letting $k = n$, we could determine such a subset (i.e., a basis for the span) of n vectors, for any n , in time $O(n^2)$. However, one can do better.

Lemma 3.2. *A maximal linearly independent subset from a set of n vectors can be found in time $O(n)$.*

Proof. We assume the dimension d to be constant, and thus, a maximal linearly independent set of d vectors can be determined in constant time.

Let $S = \{v_0, v_1, \dots, v_{n-1}\}$ be the set of n vectors from which a maximal linearly independent set must be found. Let M be a matrix of dimension $d \times d$ with rows initialized to the vectors v_0, v_1, \dots, v_{d-1} . Use row reduction, tracking the vector corresponding to each row (e.g., updating as needed upon a row exchange), to determine a linearly independent subset of these vectors. *Time* $O(1)$.

If these vectors span \mathbb{R}^d then the algorithm may terminate. Otherwise, discard the zero rows from the row reduced M , replacing them with vectors from S while still maintaining a correspondence between rows and vectors. *Time* $O(1)$.

Begin the next iteration by performing a row reduction on M to find a linearly independent set. (Note that one needs only to reduce the rows of M that were most recently added.) After the row reduction on each iteration, zero rows are replaced with vectors from S . The iterating continues until either S is

exhausted or a linearly independent set of size d is found. This repeats at most $n - d$ times. *Time* $O(n)$.

The resulting set of vectors forms a maximal linearly independent set. \square

The following lemma demonstrates the strength of the hypothesis needed by our algorithm.

Lemma 3.3. *Given that no d hyperplanes of a set in \mathbb{R}^d pass through the same line, no k hyperplanes of that set contain the same $(d - k + 1)$ -flat for $3 \leq k \leq d$.*

Proof. Suppose hyperplanes h_0, h_1, \dots, h_{k-1} all contain a $(d - k + 1)$ -flat, then $\dim(h_0 \cap h_1 \cap \dots \cap h_{k-1}) \geq d - k + 1$. Thus, by intersecting with an additional $(d - k)$ hyperplanes, for a total of d hyperplanes, (assuming no two are parallel) the resulting flat will have dimension at least $(d - k + 1) - (d - k) = 1$, and $\dim(h_0 \cap h_1 \cap \dots \cap h_{d-1}) \geq 1$, i.e., a contradiction. (If any two hyperplanes in the intersection are parallel, the result is an empty set.) \square

We will assume that the intersection of two flats can be found in constant time.

Theorem 3.4. *Given an arrangement of n hyperplanes in \mathbb{R}^d , not all parallel, not all passing through the same point, and no d passing through a line, there exists an algorithm to find an ordinary intersection point, or determine that none exists, in time $O(n \log n)$.*

Proof. Let H be the set of n hyperplanes, $\{h_0, h_1, \dots, h_{n-1}\}$, in \mathbb{R}^d , $n \geq d$, not all parallel, no d passing through a line and not all passing through the same point.

For each hyperplane, compute its normalized normal vector. The first non-zero coordinate of each normal vector should be positive (replacing the vector by its negative if necessary), so that two hyperplanes are parallel if and only if their normal vectors are the same. *Time:* $O(n)$.

Sort the hyperplanes, lexicographically, by their normals into k families of parallel hyperplanes. That is, let $H^{(0)}, H^{(1)}, \dots, H^{(k-1)}$ each be a set of hyperplanes such that for any two $h \in H^{(i)}, h' \in H^{(j)}$, h and h' are parallel if and only if $i = j$. Since parallelism is an equivalence relation, this forms a partition on the set H . So, obviously, $|H^{(0)}| + |H^{(1)}| + \dots + |H^{(k-1)}| = |H| = n$. Let $h_i^{(j)}$, for $0 \leq i < |H^{(j)}|$, be the members of the set $H^{(j)}$. *Time: $O(n \log n)$.*

For each set of hyperplanes, $H^{(i)}$, there is a distinct normal. Use the algorithm described in Lemma 3.2 to find a maximal linearly independent set from these normal vectors, tracking for each normal the associated hyperplane family. If the maximal linearly independent set does not span \mathbb{R}^d , then by Lemma 3.1 there will exist no intersection point, and the algorithm is finished *Time: $O(n)$.*

From now on we assume that an intersection exists, and therefore the number of hyperplane families, k , is at least d .

Let M be the plane formed by intersecting a member from each of the first $d - 2$ sets of hyperplanes from the d sets whose normals formed the basis in the previous step. Without loss of generality, we will assume that these $d - 2$ families are H^0, H^1, \dots, H^{d-3} . Thus, $M = h_0^{(0)} \cap h_0^{(1)} \cap \dots \cap h_0^{(d-3)}$. By Lemma 3.3, $\dim(M) = 2$. *Time to determine M : $O(n)$.*

Let L be the lines formed by intersecting each of the remaining hyperplane families with M , i.e., $L = \{l_i = M \cap h_i : h_i \in (H^{(d-2)} \cup H^{(d-1)} \cup \dots \cup H^{(k-1)})\}$, where each l_i is a line. (Note that the hyperplanes used to form lines on M all intersect M since they are not parallel to any of the hyperplanes used in the construction of M .) *Time to determine the set L : $O(n)$.*

Consider the following cases.

Case 1: The lines of L are all parallel on M .

Consider these hyperplanes in projective space. There exists a point on the hyperplane at infinity, p_∞ , that is incident to all lines of L , and thus, incident to

all hyperplanes in $\{H^{(d-2)}, H^{(d-1)}, \dots, H^{(k-1)}\}$. Furthermore, p_∞ is incident to all hyperplanes in $\{H^{(0)}, H^{(1)}, \dots, H^{(d-3)}\}$, since M and all of its constituent hyperplanes, and thus, the parallel hyperplane families, pass through p_∞ . Since the hyperplanes of H all share a common point at infinity, there will be no finite intersection point. However, since we showed that the hyperplane normals spanned \mathbb{R}^d , this case is not possible.

Case 2: The lines of L are all concurrent on M , and there are at least three of them.

For this case, we know that $|H^{(d-2)}| = |H^{(d-1)}| = \dots = |H^{(k-1)}| = 1$. Let p be the point on M at which the lines of L all cross. *Time to determine if all concurrent: $O(n)$.*

Since the lines of L all pass through p , we will construct another plane M' (parallel to M) by using one alternative member from one of the first $d - 2$ parallel families.

If no alternative member exists, then all hyperplane families contain just one member. Thus, all hyperplanes pass through point p , in violation of the hypothesis. In this case, no ordinary point exists and the algorithm terminates.

Without loss of generality, assume $h_1^{(0)}$ (i.e., a second member from the set $H^{(0)}$) is the alternative member used to construct M' . That is, let $M' = h_1^{(0)} \cap h_0^{(1)} \cap \dots \cap h_0^{(d-3)}$. *Time to construct M' : $O(n)$.*

Construct the set L' in an analogous manner to the construction of L , i.e., let $L' = \{l'_i = M' \cap h_i : h_i \in (H^{(d-2)} \cup H^{(d-1)} \cup \dots \cup H^{(k-1)})\}$. The lines of L' cannot all be parallel by the same argument used in Case 1.

Assume the lines of L' are again all concurrent at a finite point p' . Then the line determined by p and p' , i.e. $\overline{pp'}$, is contained in the $d - 3$ hyperplanes used to construct both M and M' , which excludes the two hyperplanes used from the set $H^{(0)}$. (An intersection of hyperplanes containing two distinct

points, also contains the line that connects them.) That is, $\overline{pp'}$ is contained in $h_0^{(1)}, h_0^{(2)}, \dots, h_0^{(d-3)}$. The line $\overline{pp'}$ is also contained in the three or more hyperplanes that formed the lines of L and L' . Altogether, there must be at least $(d-3)+3 = d$ hyperplanes containing the line $\overline{pp'}$, in violation of our hypothesis.

Time to determine if all concurrent on M' : $O(n)$.

Therefore, M' contains an ordinary intersection, and we may proceed to the next case, *mutatis mutandis*.

Case 3: The lines of L form an ordinary intersection on M .

The ordinary intersection formed by the lines of L can be found using the algorithm given in Section 2.3. Assume l_i and l_j form an ordinary intersection on M . This point is the intersection of the hyperplanes $h_0^{(0)} \cap h_0^{(1)} \cap \dots \cap h_0^{(d-3)}$ and exactly two other hyperplanes (which formed l_i and l_j), and thus, at the intersection of d hyperplanes. Therefore, we have found an ordinary intersection.

Time: $O(n \log n)$. □

4 Arrangements of Pseudolines

4.1 Ordinary Intersection Points

Now consider an arrangement of pseudolines, any two of which cross and not all at the same point. Any such arrangement contains an ordinary intersection, and the best result in this area is that of Csimá and Sawyer [4], who extended their $\frac{6n}{13}$ result to also include ordinary intersections among pseudolines in the projective plane. An elegant proof of the existence of ordinary intersections can be found using Euler's formula to find an inequality due to Melchior [14]. See Felsner's book [7] for excellent coverage of this and other results related to Sylvester's Problem.

Arrangements of pseudolines, as discussed in this paper, have certain prop-

erties that are assumed:

- Each pseudoline goes off to infinity in both directions.
- No pseudoline crosses itself.
- Each pair of pseudolines intersects at exactly one point, and at that point cross.
- More than two pseudolines may cross at a single point (otherwise the intersection is ordinary).
- The pseudolines do not all cross at the same point (i.e., there is more than one intersection point).

See [7] for a more complete explanation of pseudoline arrangements and their properties.

It is assumed that given a point P and a pseudoline L , one can determine whether P lies on L in time $O(1)$. Therefore, in an arrangement of n pseudolines, the pseudolines that cross P can be determined in time $O(n)$. It is also assumed that the intersection point of any two pseudolines can be found in time $O(1)$.

Recently a couple of results related to the following algorithm have been published or submitted. Pretorius and Swanepoel in [18] provide proof of a theorem that generalizes both Sylvester-Gallai and Motzkin-Rabin. Their proof utilizes a sequence of successively smaller triangles that terminates with finding the desired intersection point. A similar method is also used by Lenchner in [13]. Note that one might also see similarity between these proofs and Motzkin's as published in [6] (i.e. they utilize what Motzkin calls "characteristic triangles").

The algorithm described below was inspired by the recent proof given by Lenchner in [13]. This algorithm can be used to find ordinary point in an arrangement of lines, and by duality an ordinary line determined by a set of

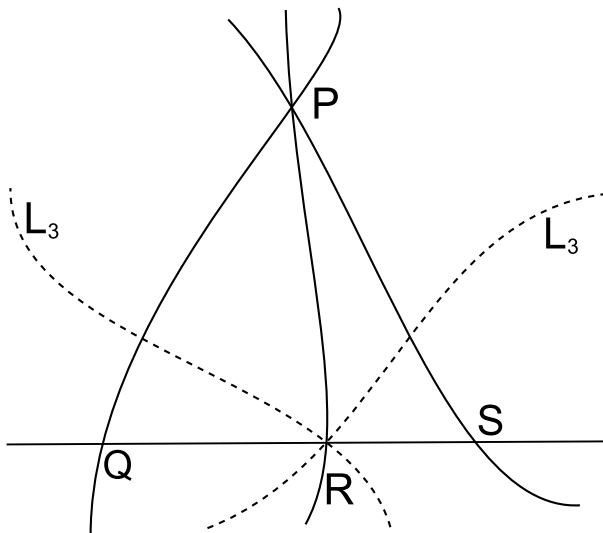


Figure 3: If point R is not ordinary, then a third pseudoline L_3 must cross either segment PQ or PS .

points. We must also mention that a $O(n^2)$ algorithm could be obtained by an incremental construction of the arrangement that tracks the intersections that are created. However, such an algorithm would not also prove the existence of an ordinary intersection point.

4.2 Algorithm to Find an Ordinary Point in Time $O(n^2)$

Theorem 4.1. *An ordinary intersection can be found in an arrangement of pseudolines in time $O(n^2)$.*

Proof. Let L_0 , L_1 and L_2 be any three pseudolines of arrangement \mathcal{A} that intersect at three distinct points. *Time to find three such lines: $O(n)$.*

Let P be the intersection point of L_1 and L_2 . If P is ordinary, then the algorithm is done. *Time to determine whether P is ordinary: $O(n)$.*

Otherwise, there are at least three pseudolines (L'_1, L'_2 and L'_3) crossing at P . Consider L_0 to be “horizontal” and P “above” L_0 . Let points Q , R and S

be the points of intersection left to right on L_0 of the three pseudolines crossing at P (i.e., L'_1, L'_2 and L'_3). If R is an ordinary intersection, then the algorithm is done. *Time to determine whether R is ordinary: $O(n)$.*

Otherwise there is a pseudoline, L_3 , crossing at R that either crosses the finite segment QP or PS . *Time to determine where L_3 crosses: $O(1)$.*

Without loss of generality, assume this pseudoline crossing R also crosses QP . This pseudoline is defined to be the triangle's *dividing line*. The configuration will now be reoriented for recursion, letting R be the intersection of L_3 with pseudoline QP , P the previous R , Q the previous P , and S the previous Q . *Time to reorient the configuration for recursion: $O(1)$.*

The following lemma states that this recursion repeats no more than n times, yielding a time $O(n^2)$ algorithm. \square

Lemma 4.2. *No pseudoline is used by the algorithm as a dividing line more than once.*

Proof. Each *dividing line* L crosses the interior of a triangle, dividing it into two parts. All subsequent *dividing lines* used by the algorithm must cross the interior of one of those parts, of which L lies on the boundary. \square

This second algorithm has a potential advantage over our first since it may stop early, possibly at the first intersection P (i.e. time $\Omega(n)$). By duality, this algorithm also can be used to find ordinary lines in an arrangement of points, with the same time complexity.

4.3 Existence of Monochromatic Points in a Bichromatic Arrangement

In a bichromatic arrangement of pseudolines, any two crossing and not all crossing at the same point, a monochromatic intersection point always exists, but it

might not exist for both colors. An arrangement containing monochromatic intersections of only one color is called “biased” (see [8]). The existence of biased arrangements requires any algorithm in search of a monochromatic intersection to consider both colors (or at least be run twice if limited to a specific color).

The previous algorithm will now be modified to find a monochromatic intersection. While Chakerian [2] and others have proven that lines in the real projective plane always determine a monochromatic intersection (and an argument similar to theirs might be extended to include pseudolines), the present authors are unaware of a proof that explicitly extends this result to pseudolines in the euclidean plane. The algorithm below provides such a proof.

In [17], Pretorius and Swanepole provide an algorithm (i.e. an algorithmic proof) to find a monochromatic line in a bichromatic set of points, apparently in time $O(n^2)$ (although the present authors are unaware of a “worst-case” instance for their algorithm). Note that the bichromatic pseudoline problem is more general than that of points, since not every pseudoline arrangement has a dual.

As with the previous algorithm, it is assumed that given a point P and a pseudoline L , one can determine whether P lies on L in time $O(1)$. Therefore, in an arrangement of n pseudolines, the pseudolines that cross P can be determined in time $O(n)$. It is also assumed that the intersection point of any two pseudolines can be found in time $O(1)$.

4.4 Algorithm to Find a Monochromatic Intersection in a Bichromatic Arrangement of Pseudolines

Theorem 4.3. *A monochromatic intersection in a bichromatic arrangement of pseudolines may be found in time $O(n^2)$.*

Proof. Let L_0 be a pseudoline from an arrangement, \mathcal{A} , containing n pseudolines

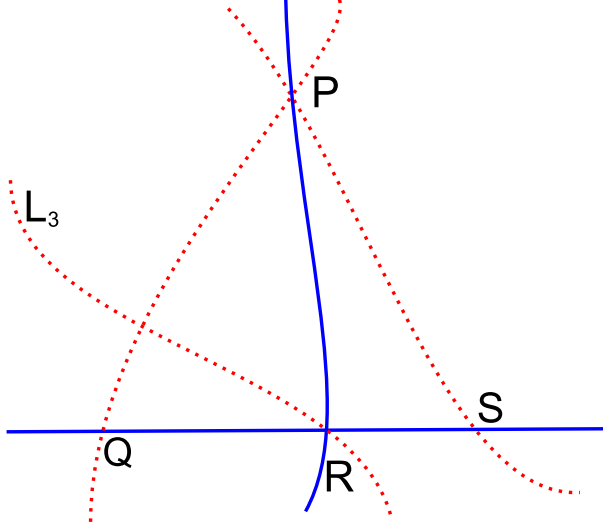


Figure 4: If point R is monochromatic, then a third pseudoline L_3 with a different color must cross either segment PQ or PS .

each colored one of red or blue, any two of which cross but not all cross at the same point. Consider L_0 to be “horizontal” and, without loss of generality, assume its color is blue. *Time: $O(1)$.*

Let Q and S be the leftmost and rightmost intersection points on L_0 . Assume a red pseudoline crosses at Q , another red pseudoline crosses at S , and let P be their intersection point “above” L_0 . If this assumption is false (i.e. red pseudolines do not cross both Q and S), then at least one of Q or S is monochromatic and the algorithm is done. *Time to find Q and S and determine whether they are monochromatic: $O(n)$.*

If P , the intersection of the red pseudolines crossing Q and S , is monochromatic then again, the algorithm is done. Otherwise, a blue pseudoline L_2 crosses P and intersects L_0 at point R , between Q and S . See Figure 4.

At this point, the “setup” is complete and we begin the first step of a (potentially) recursive process to find a monochromatic intersection.

If R it is monochromatic (blue), then the algorithm is done. *Time to deter-*

mine whether R is monochromatic: $O(n)$.

Otherwise, a red pseudoline, L_3 , crosses R and intersects either segment PQ or segment PS . Without loss of generality, assume it crosses PQ . This pseudoline is defined to be the triangle's (i.e. $\triangle PQR$'s) *dividing line*. The configuration will now be reoriented for recursion, letting R be the intersection of L_3 with pseudoline QP , P the previous R , Q the previous P , and S the previous Q . *Time to reorient the configuration for recursion: $O(1)$.*

Note that each step of the recursive process, expects R to possess a different, possibly monochromatic, color. So for the first and all other odd numbered steps it would expect “blue”, and likewise “red” for the even. Again, we refer to Lemma 4.2 to show that this algorithm runs in time $O(n^2)$. \square

5 Conclusion

It is conjectured that both $O(n \log n)$ algorithms presented here are within a constant factor of the best upper bound for time.

It would be interesting to know whether an algorithm to find an ordinary intersection in an arrangement of pseudolines could also perform in time $O(n \log n)$. Likewise, it would be interesting to know whether an algorithm to find a monochromatic intersection in a bichromatic arrangement of pseudolines (or even lines) could perform in time $O(n \log n)$.

References

- [1] W. Bonnice and L. M. Kelly, *On the number of ordinary planes*, Journal of Combinatorial Theory, Series A **11** (1971), no. 1, 45–53.
- [2] G. D. Chakerian, *Sylvester's problem on collinear points and a relative*, The American Mathematical Monthly **77** (1970), no. 2, 164–167.

- [3] H. S. M. Coxeter, *A problem of collinear points*, The American Mathematical Monthly **55** (1948), no. 1, 26–28.
- [4] J. Csima and E. Sawyer, *There exist $6n/13$ ordinary points*, Discrete and Computational Geometry **9** (1993), no. 1, 187–202.
- [5] G.A. Dirac, *Collinearity properties of sets of points*, Quarterly Journal of Mathematics **2** (1951), no. 1, 221–227.
- [6] Paul Erdős and George Purdy, *Extremal problems in combinatorial geometry*, Handbook of Combinatorics (R. Graham, M. Grötschel, and L. Lovász, eds.), vol. 1, Elsevier, Amsterdam, 1995, pp. 809–873.
- [7] Stefan Felsner, *Geometric graphs and arrangements : some chapters from combinatorial geometry*, 1st ed., Vieweg, Wiesbaden, 2004.
- [8] Branko Grünbaum, *Monochromatic intersection points in families of colored lines*, Geombinatorics **9** (1999), no. 1, 3–9.
- [9] Sten Hansen, *A generalization of a theorem of Sylvester on the lines determined by a finite point set*, Mathematica Scandinavica **16** (1965), 175–180.
- [10] L.M. Kelly and W.O.J. Moser, *On the number of ordinary lines determined by n points*, Canadian Journal of Mathematics **10** (1958), no. 2, 210–219.
- [11] Jonathan Lenchner, *On the dual and sharpened dual of Sylvester’s theorem in the plane*, IBM Research Report **RC23411** (2004), no. W0409-066.
- [12] ———, *An improved bound for the affine Sylvester problem*, 19th Canadian Conference on Computational Geometry (CCCG) (Ottawa, ON), 2007, pp. 57–60.
- [13] ———, *A new proof of the Sylvester-Gallai theorem*, IBM Research Report **RC24603** (2008), no. W0807-057.

- [14] E. Melchior, *Über Vielseite der projektiven Ebene*, Deutsche Math. **5** (1941), 461–475. MR MR0004476 (3,13c)
- [15] Asish Mukhopadhyay, Alok Agrawal, and Ravi Mohan Hosabettu, *On the ordinary line problem in computational geometry*, Nordic J. of Computing **4** (1997), no. 4, 330–341, 640123.
- [16] Asish Mukhopadhyay and Eugene Greene, *The ordinary line problem revisited*, 19th Canadian Conference on Computational Geometry (CCCG) (Ottawa, ON), 2007, pp. 61–64.
- [17] Lou M. Pretorius and Konrad J. Swanepoel, *An algorithmic proof of the motzkin-rabin theorem on monochrome lines*, The American Mathematical Monthly **111** (2004), no. 3, 245–251.
- [18] ———, *The sylvester-gallai theorem, colourings and algebra*, Discrete Mathematics **309** (2009), no. 2, 385–399.
- [19] George B. Purdy and Justin W. Smith, *Lines, circles, planes and spheres*, 2009.
- [20] Robert Steinberg, *Solution for problem 4065*, The American Mathematical Monthly **51** (1944), no. 3, 169–171.
- [21] James Joseph Sylvester, *Mathematical question 11851*, Educational Times **59** (1893), 98–99.